Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Multi-level cross-layer protocol for end-to-end delay provisioning in wireless multimedia sensor networks*

Hossein HADADIAN NEJAD YOUSEFI, Yousef SEIFI KAVIAN†‡, Alimorad MAHMOUDI

Electrical Engineering Department, Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz 6135783151, Iran

†E-mail: y.s.kavian@scu.ac.ir

Received Dec. 18, 2017; Revision accepted July 2, 2018; Crosschecked June 11, 2019

Abstract: Rapid developments in information and communication technology in recent years have posed a significant challenge in wireless multimedia sensor networks (WMSNs). End-to-end delay and reliability are the critical issues in multimedia applications of sensor networks. In this paper we provide a new cross-layer approach for provisioning the end-to-end delay of the network at a desirable level of the packet delivery ratio (PDR), used here as a measure of network reliability. In the proposed multi-level cross-layer (MLCL) protocol, the number of hops away from the sink is used to set a level for each node. A packet is routed through the path with the minimum hop count to the sink using this level setting. The proposed protocol uses cross-layer properties between the network and medium access control (MAC) layers to estimate the minimum delay, with which a node can deliver a packet to the sink. When a node wants to send a packet, the MLCL protocol compares this minimum delay with the time to live (TTL) of a packet. If the TTL of the packet is higher than the minimum delay, the node sends the packet through the path with the minimum delay; otherwise, the node drops the packet as the node cannot deliver it to the sink within the TTL duration. This packet dropping improves network performance because the node can send a useful packet instead of an unusable packet. The results show a superior performance in terms of end-to-end delay and reliability for the proposed protocol compared to state-of-the-art protocols.

Key words: Wireless multimedia sensor networks; Cross layering; Time to live; End-to-end delay; Quality of service
https://doi.org/10.1631/FITEE.1700855 **CLC number:** TP393

1 Introduction

In recent years, wireless sensor networks (WSNs) have been widely used in many areas (Akyildiz et al., 2006). WSNs consist of sensor nodes deployed in an area to sense and send data to a base station. With recent developments in low-cost hardware such as complementary metal-oxide-semiconductor (CMOS) cameras, microphones, and passive infrared (PIR) sensors, wireless nodes can be equipped with these modules to facilitate the operation of unmanned monitoring systems (Akyildiz et al., 2008). Wireless nodes equipped with these peripherals in wireless multimedia sensor networks (WMSNs) must be designed to handle a higher data rate than conventional networks.

Wireless multimedia sensor nodes gather a large and widely distributed volume of sensed data. The nodes encounter many limitations in gathering this volume of data, including the wireless channel, bandwidth limitation, energy consumption, security, and decentralized management (Dargie and Poellabauer, 2010). These limitations also make it difficult to design WMSNs to satisfy network requirements. End-to-end delay provisioning and reliability are the most critical requirements in multimedia applications. An efficient design of the medium access control (MAC) and network layers is important for meeting these requirements.

---

‡ Corresponding author

* Project supported by the Shahid Chamran Univesity of Ahvaz (No. 96/3/02/16670)

ORCID: Yousef SEIFI KAVIAN, http://orcid.org/0000-0002-2076-4927

Many layered protocols have been proposed to overcome these challenges in conventional layered networks (Yigitel et al., 2011; Ehsan and Hamdaoui, 2012; Kumar et al., 2012). However, it has been shown that cross-layer designs give better performance than layered counterpart designs (Mendes and Rodrigues, 2011). Cross-layer approaches have been designed for a variety of purposes including energy efficient communication (Lin et al., 2009), resource allocation (Feng et al., 2014), optimum routing (Messaoudi et al., 2017; Li et al., 2018), congestion control (Wang et al., 2007), and fault tolerance management (Karaca and Sokullu, 2012).

In recent years, some services have been developed that need strict deadline constraints in many applications such as real-time control systems and telecommunication systems with voice and video traffic. In these systems, packets must be delivered to the destination before their deadlines are reached. Otherwise, they become unusable and the node drops the packet to improve network performance. To reduce the number of dropped packets, these systems use priority queue scheduling algorithms. A system with earliest deadline first (EDF) scheduling can deliver more usable packets than the familiar first-in-first-out (FIFO) scheduling. EDF scheduling is optimal for this purpose (Kruk et al., 2011). Thus, our proposed protocol schedules the packets according to EDF scheduling. Consequently, the node sends the packet with the shortest time to live (TTL). Also, when a node detects that it cannot deliver the packet before the deadline, the packet is dropped before the deadline is reached. The sensor nodes make this detection by estimating the minimum delay time (MDT), with which they can deliver a packet to the destination.

We propose a multi-level cross-layer (MLCL) design to meet reliability and end-to-end delay requirements. The MLCL protocol combines the functionality of the network and MAC layers, and has two phases. In the first phase, the MLCL protocol divides the network into several levels. The level for each node determines the minimum hop count to the sink node. Each node acquires its level in the first phase. After the first phase, the node knows the number of hops to the sink node. Therefore, the node forwards its DATA packet to a neighbor node whose level is lower than its own. In the second phase, each node estimates the average delay, with which it can deliver a packet to the sink through each lower level neighbor. Then, it sends the packet to the neighbor with the lowest average delay.

2 Related work

WMSN constraints have prompted many initiatives to find solutions in a variety of applications. Providing a throughput comparison for network efficiency is a crucial task. The primary task of WMSNs must be transparent to achieve an appropriate comparison between protocols. For instance, throughput maximization for event-based applications is not as important as it is for monitoring applications. The two most important challenges for real-time systems are end-to-end delay and quality of service (QoS) reliability for packet deliveries. Satisfying these QoS requirements is difficult for two reasons: (1) Wireless sensor nodes may require multi-hop transmissions to reach the sink; (2) Some wireless transmissions may be unsuccessful (Hou, 2015).

Although Akyildiz et al. (2006) and Misra et al. (2008) have discussed and classified the challenges existing at each layer of WMSNs, they have only briefly investigated cross-layer approaches to overcome the challenges and guarantee QoS parameters. Hamid and Hussain (2014) gave an overview of QoS parameters that can be satisfied by each layer. They also discussed in detail the existing cross-layer approaches and categorized the significant gains that can be achieved through cross-layer interaction in WMSNs.

The XLP design (Vuran and Akyildiz, 2010) is a cross-layer protocol proposed for efficient and reliable communication with low energy consumption and local control of congestion. The functionality of this cross-layer is a combination of the functionality of all layers in the open system interconnection (OSI) model. The concept of initiative determination is introduced for easy implementation. The initiative determination consists of four binary terms on the node energy, link status, arrival packet rate, and occupancy level. A request-to-send/clear-to-send (RTS/CTS) mechanism is used for data transmission. A node receives an RTS packet. If the node is closer to the sink node than the RTS packet sender and the initiative determination is confirmed, the node sends a CTS packet. Otherwise, the node goes to sleep mode for one transmission period. A cross-layer design has been used by Singh and Verma (2017) for energy efficient routing, and another was used by Al-Wazedi and Elhakeem (2011) to assign a cluster head using the concept of weighted probability.

In WMSNs, due to resource constraints, cross-layer designs are introduced to optimize the trade-off between QoS and resource cost. Therefore, some protocols join the network and MAC layers to meet end-to-end delay requirements (Felemban et al., 2006; Sahoo and Chilukuri, 2010; Hamid and Bashir, 2013; Demir et al., 2014). Other protocols use the interaction between the application and network layers (Bhuiyan et al., 2011; Lin and van der Schaar, 2011). Wang HG et al. (2008, 2009) presented a cross-layer design for distributed source coding applications in WSNs, in which coding rate allocation in the application layer and link assignment in the routing layer are jointly designed. Wang W et al. (2009) proposed a joint design of resource allocation at the link layer and physical layer with the rate distributions at the application layer, for multimedia transmission over WSNs.

The MMSPEED protocol, which uses ReInforM (Deb et al., 2003) and SPEED (He et al., 2003) protocols, was proposed to provide differentiated QoS options in timeliness and reliability domains (Felemban et al., 2006). The nodes maintain the immediate neighbor information such as the distance and delay to each neighbor node. Each node uses the information to estimate the progress speed of each immediate neighbor node for localized geographic routing. If these estimated values are lower than the progress speed required to achieve the end-to-end delay limitation, the node drops the packet. Otherwise, it forwards the packet to the node whose progress speed is higher than what is required. Each node evaluates the total reachable probability. The node using this evaluation forwards the packet to multipath to satisfy reliability requirements.

QoSMOS is a QoS architecture which unifies the network and medium access control layers into a single cross-layer module called XLCM (Demir et al., 2011, 2014). The XLCM module is able to provide various QoS levels of soft delay, reliability, and throughput. This module unifies the network and MAC functionality by assuming that the localized geographical routing uses only the information of the immediate neighbors. Therefore, it uses local information to eliminate local congestion.

The XL-WMSN cross-layer protocol unites an energy-aware admission control, a delay- and traffic-aware routing protocol, and an end-to-end deadline-aware duty cycle (Hamid and Bashir, 2013). The end-to-end deadline-aware duty cycle provides a delay constrained delivery of multimedia data while conserving energy. It is assumed that the wireless sensor nodes are deployed in a grid-like arrangement in the whole network. The XL-nodes estimate the packet service time and channel utilization to establish a path.

3 MLCL architecture

MLCL architecture is proposed to satisfy the end-to-end delay and reliability requirements. The end-to-end delay provisioning and reliability of each packet depend on the specific network requirements. In this architecture, the end-to-end delay requirement is indicated by a TTL field in each packet. In other words, the TTL field indicates the time that the packet should take to be delivered to the destination. When the node creates the packet, it inserts the corresponding TTL into the TTL field of the packet. The TTL value is updated according to the time stayed in nodes along the route.

The functionality of MLCL is a combination of the functionality of the network and MAC layers to deliver a packet to the destination before its TTL reaches zero. The MLCL architecture consists of several components (Fig. 1).

The cross-layer service provider (CLSP) manages and controls all components of the MLCL architecture. Using the TTL timer, the CLSP measures how long a packet has stayed in the node. Thus, the TTL value of all packets should be updated if the TTL timer is reset. The TTL updater module updates the TTL field by subtracting the TTL field value from the TTL timer value.

The CLSP has two main functions: Insert and Select-Send. The Insert function illustrated in Algorithm 1 is called when a packet is received from either

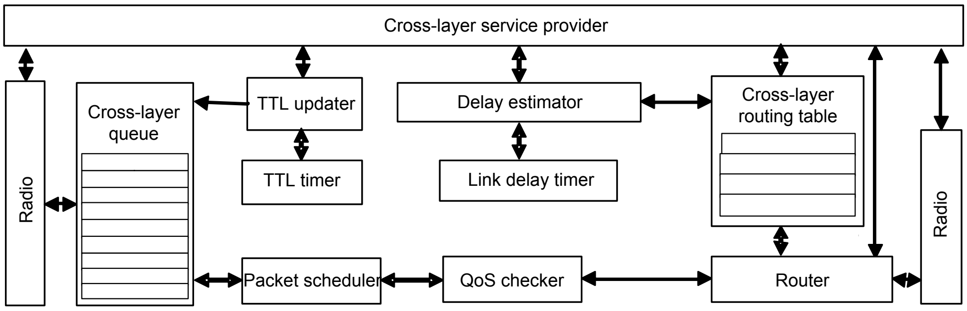

Fig. 1 The multi-level cross-layer (MLCL) architecture

radio or an application layer. The Select-Send function illustrated in Algorithm 2 is called when at least one packet is in the cross-layer queue.

Algorithm 1 Insert function

```
Update the TTL field of all packets in the cross-layer queue
Reset the TTL timer
Insert the received packet into the cross-layer queue
```

Algorithm 2 Select-Send function

```
Get the packet with the minimum TTL value from the cross-layer queue
Get the minimum idle from the delay estimator module
if TL value < min(MDT) then
    Drop the packet
else
    Select the neighbor with the minimum MDT for the next hop
    Update the TTL field of the packet
    Send the packet with the proposed RTS/CTS mechanism
end if
```

4 MLCL protocol

The MLCL protocol is divided into two phases: an initial phase and a QoS provisioning phase. We used the initial phase in our previous work (Hadadian and Kavian, 2016) to avoid congestion and achieve a uniform traffic distribution. After the initial phase, all nodes set their level to be applied (Pandya and Mehta, 2012; Huang et al., 2013). The nodes neighboring the sink set their level to "one" because they can send their data directly to the sink node. The neighboring "level-one" nodes that are not neighboring the sink set their level to "two." The level-two nodes know they are one hop away from the sink node. All nodes find their level likewise.

The MLCL protocol uses a routing table and the initial phase to send data to the sink. Nodes with a lower level are listed in the routing table. Thus, a node transmits the data to the lower level node. This flow continues until the data are received at the sink node.

4.1 Initial phase

In the initial phase, the sink node starts the network leveling by setting level-zero and sending a get-level packet. This packet comprises the address and level of the sender. If the receiver of the get-level packet has not yet set its level or its level is higher than the level inserted in the get-level packet, it sets its level equal to the level inserted in the get-level packet plus one, and sends the get-level packet. After the initial phase, all nodes find the minimum hop count away from the sink node and insert the address of lower level nodes in the routing table. An example of network leveling is illustrated in Fig. 2. It is possible that a node could not set a level because of a collision. Therefore, when a node detects a collision, it sends a set-level packet to inform the neighboring nodes to send the get-level packet again.

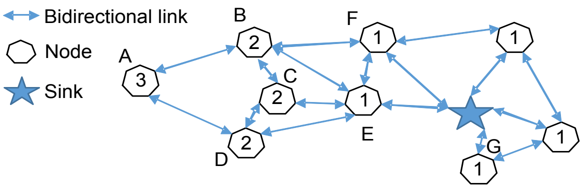

Fig. 2 The level of each node after the initial phase

4.2 QoS provisioning phase

When a node has a packet in the cross-layer queue and decides to send the packet to the sink node, the CLSP first checks the wireless channel status. If the channel is idle for a specific period, the node transmits an RTS packet, and the countdown of its backoff timer starts waiting to receive a CTS packet. The period is equal to the sum of the RTS and CTS packet durations. The RTS is sent to a node that has the lowest MDT in the routing table. This node chooses to use the delay estimator component. When the selected node receives the RTS packet, it inserts the address and level of the sender and receiver into a CTS packet and sends this packet. After receiving the CTS packet, the RTS sender updates the TTL field of the DATA packet, transmits it, and stops the backoff timer. When the CTS sender has received the DATA packet, it sends an ACK packet. Otherwise, if the CTS sender does not receive the DATA packet, it sends an NACK packet. The address of the next hop is inserted into the ACK packet, and the ACK packet can role as the RTS packet. When the node receives the NACK packet, it sends the DATA packet again. However, when the node does not receive ACK or NACK packets, it again performs RTS/CTS handshaking.

For energy saving, when a node other than the selected receiver receives an RTS packet, it goes into sleep mode until one transmission ends. The CTS packet receiver, except the RTS packet sender, goes to sleep until one transmission ends, if its level is lower than that of the RTS packet sender. Otherwise, it goes into sleep mode for the duration of two transmissions. The node chooses this sleep duration to save energy because it cannot participate in transmitting the DATA packet that the CTS sender must relay directly after this transmission ends. An example of the timing protocol applied in Fig. 2 is shown in Fig. 3 when node A has a DATA packet to send to the sink.

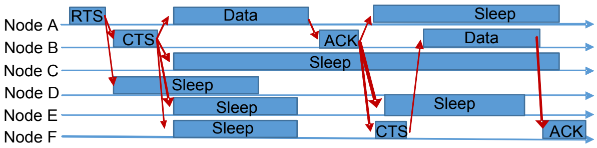

Fig. 3 The proposed RTC/CTS handshaking example of Fig. 2

To select the next hop with the minimum delay, the delay estimator module uses the minimum delay approximation algorithm to estimate the MDT at each node within which it can deliver a packet to the sink.

4.3 Minimum delay approximation algorithm

The node measures link delay time (LDT), which is the interval between the time when it decides to get the channel and the time of successful data transmission. When a transmission fails, the node tries again to get the channel, and the LDT counts until a successful transmission is achieved. Therefore, LDT is a good indicator of delay, congestion, and utilization of the link (He et al., 2003). After that, the node calculates the exponential weighted moving average of the LDT:

$$\mathrm{LDT}_{\mathrm{AVG}}=\alpha\cdot\mathrm{LDT}(n)+(1-\alpha)\mathrm{LDT}_{\mathrm{AVG}}, \tag{1}$$

where α ($0\le\alpha\le1$) is a constant value used to regularize the weights to the current LDT measurements with respect to past measurements. A higher α can make the algorithm more robust to a sudden increase and congest the burst of data.

The one-level nodes estimate the MDT and insert it into the RTS and CTS packets. The MDT for one-level nodes is defined as follows:

$$\mathrm{MDT}_1=\mathrm{LDT}_{\mathrm{AVG}}(\mathrm{QL}+1), \tag{2}$$

where QL is the number of packets in the queue. We add one to the QL because when the cross-layer queue is empty, the MDT cannot be zero. If we do not add one, it is possible that a link is not reliable, but its MDT is zero. Other nodes find the minimum of the MDTs that are received from lower level nodes and calculate their MDTs as follows:

$$\mathrm{MDT}_i=\mathrm{LDT}_{\mathrm{AVG}}(\mathrm{QL}+1)+\min(\mathrm{MDT}_{i-1}), \tag{3}$$

where i determines the node's level. Each node inserts its MDT in the RTS and CTS packets. When a node receives RTS or CTS packets, the node extracts the MDT and sender address from the packet and saves them to the routing table. The calculated MDT is the minimum delay time during which the node can deliver a packet to the sink node. The MDT is an excellent factor indicating congestion along the route to the sink node. For instance, consider this algorithm applied in the network of Fig. 2. Assume the length of all cross-layer queues is one and the LDT average values of nodes A, B, C, D, E, and F are equal to 3, 5, 6, 5, 7, and 4, respectively. The MDT of each node is obtained by Eqs. (1)–(3) and inserted into the routing table (Table 1). These calculations are illustrated in Eq. (4):

$$\begin{cases} \mathrm{MDT_F} = 2\times 4 = 8, \\ \mathrm{MDT_E} = 2\times 7 = 14, \\ \mathrm{MDT_D} = 2\times 5 + \mathrm{MDT_E} = 24, \\ \mathrm{MDT_C} = 2\times 6 + \mathrm{MDT_E} = 26, \\ \mathrm{MDT_B} = 2\times 5 + \min(\mathrm{MDT_F},\ \mathrm{MDT_E}) = 18, \\ \mathrm{MDT_A} = 2\times 3 + \min(\mathrm{MDT_B},\ \mathrm{MDT_D}) = 24. \end{cases} \tag{4}$$

Table 1 The cross-layer routing table of Fig. 2

A		B		C	
B	24	E	24	E	26
D	30	F	18		
D		**E**		**F**	
E	24	Sink	14	Sink	8

5 Simulation evaluation

In this section, we report the evaluation of the proposed cross-layer protocols using the MIXIM package and OMNeT++ simulator. The simulation results were averaged over 10 random network topologies, and the key simulation parameters are listed in Table 2. The evaluations were carried out in three cases. In the first case, the impact of the QoS checker on the end-to-end delay was examined by evaluating the MLCL architecture with and without this module. In the second case, the effect of the event frequency on the packet delivery ratio (PDR) was evaluated. In the final case, the MLCL protocol was compared with the protocols described in the literature.

In the first case, the network was composed of 50 sensor nodes in a 60 m×60 m area with the sink coordinates (30, 10), and two events occurred at the same time at coordinates (10, 50) and (40, 50). Each sensor node, whose distance from the event center was less than 5 m, periodically sampled the data and inserted 0.25 s into the TTL field. Then, it sended the DATA packet to the next hop. The simulation results are shown in Figs. 4 and 5 for two events whose frequency, start time, and end time were 20 Hz, 5 s, and 55 s, respectively.

Table 2 Key parameters in the simulation

Design	Parameter	Value
MLCL	RTS, CTS, ACK, NACK	24 bytes
	Radio range	15 m
	Buffer length	100
	Table route size	10
	Carrier frequency	2.412×10^9 Hz
	Data	256 bytes
	Bitrate	250 kb/s
	α	0.7
QoSMOS	Maximum retransmission	8
	($\beta_{\text{min NB}}$, $\beta_{\text{max NB}}$)	(4, 10)
	($\beta_{\text{min BE}}$, $\beta_{\text{max BE}}$)	(3, 8)
	$\beta_{\text{min CW}}$	1
	$\beta_{\text{duty-cycle}}$	1
	β_{snr}	0.2
	Time slot	
	Class A	8 μs
	Class B	32 μs
	Buffer length	
	Class A	10
	Class B	90
XL-WMSN	$\mathrm{CW_{min}}$	7
	α	0.7
	$\mathrm{CW_{max}}$	62
	C_{min}	40%
MMSPEED	Maximum retransmission	7
	$\mathrm{CW_{min}}$	
	Class A	15
	Class B	31
	$\mathrm{CW_{max}}$	
	Class A	255
	Class B	511
	Reaching probability	
	Class A	0
	Class B	5
	Speed level	
	Class A	100 m/s
	Class B	25 m/s

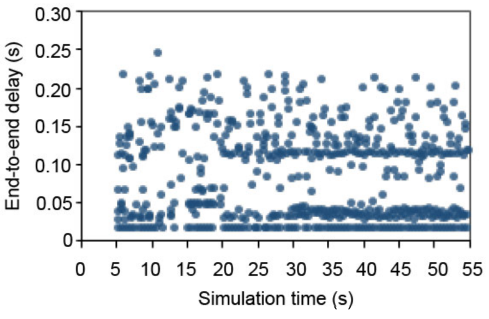

Fig. 4 The end-to-end delay of the MLCL protocol for 0.25 s end-to-end delay provisioning

Figs. 4 and 5 illustrate the end-to-end delay of each packet received by the sink versus time. In Fig. 4, the QoS checker module dropped the unusable packet. Thus, some of the other packets can be delivered with a desirable end-to-end delay. The results showed that the end-to-end delay for all received packets was less than the specified delay constraint. The simulation results in Fig. 5 showed that when the QoS checker module was not employed, the unusable packet increased the traffic load. The end-to-end delay increased and the PDR decreased with the increase of the traffic load. The PDR at the sink was 35% with, or 6% without, the QoS checker.

In the second case, the effect of the event frequency on the proposed protocol was evaluated (Fig. 6). The same network situation as used above was applied in this simulation. The results showed that when the event frequency increased, the PDR was reduced. In higher end-to-end delay provisioning, the bandwidth capacity of the wireless network caused this reduction in the PDR. Also, for a constant event frequency, degradation of the PDR occurred when low end-to-end delay provisioning was needed.

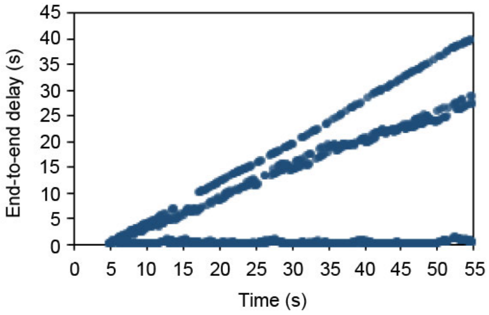

Fig. 5 End-to-end delay of the MLCL protocol without the QoS checker

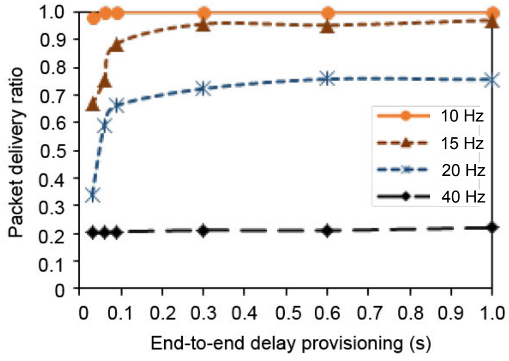

Fig. 6 Event frequency effect in the MLCL protocol

In the third case, first the proposed MLCL protocol was compared with MMSPEED (Felemban et al., 2006), QoSMOS (Demir et al., 2014), XL-WMSN (Hamid and Bashir, 2013), and XLP (Vuran and Akyildiz, 2010). Then, comparisons with MMSPEED and QoSMOS were examined in more detail.

For the first comparison, a 1 s time for the end-to-end delay provisioning was selected for one class event occurring in the network, whose conditions are described in the second case. Fig. 7 shows that the PDR decreased as the frequency of the events increased, for all protocols. When the frequency of the events increased, the average end-to-end delay increased (Fig. 8).

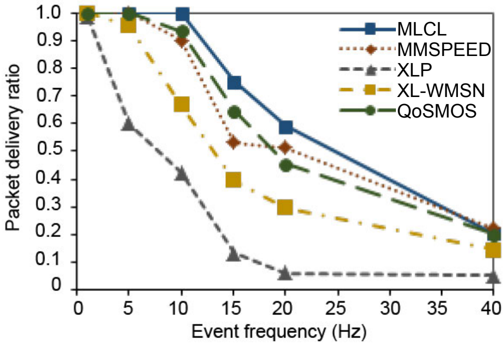

Fig. 7 Comparative packet delivery ratio (MLCL, XLP, MMSPEED, QoSMOS, and XL-WMSN)

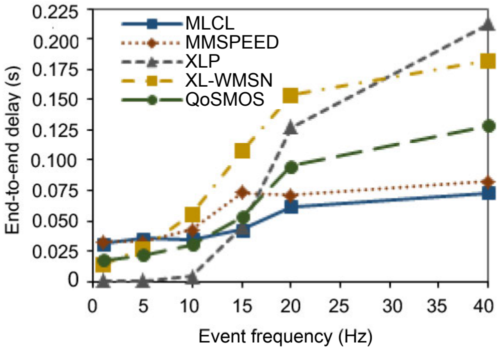

Fig. 8 Comparative end-to-end delay (MLCL, XLP, MMSPEED, QoSMOS, and XL-WMSN)

The XLP protocol is not suitable for WMSNs because when the rate of data in networks increases, congestion occurs, and thus the end-to-end delay increases. The end-to-end delay axis for the XLP protocol was scaled by 0.01 in Fig. 8. For example, for an event frequency of 20 Hz, the average end-to-end delay of the XLP protocol was 12.73 s. The PDR of the XLP protocol is the ratio of the number of packets whose delay time is less than 1 s to the total number of packets received by the sink. This value for other protocols is the ratio of the total number of packets received by the sink to the total number of packets generated by the source nodes.

XL-WMSN showed weaker performance than the rest of the protocols. XL-WMSN uses a route request mechanism to establish the path to the sink. The protocol was proposed for use in a grid network and it seemed that it did not perform well in these network situations.

For all protocols, when the network cannot well support the data rate, the collision probability and congestion level were increased through the network. Thus, the end-to-end delay increased, and the PDR decreased severely. However, the proposed protocol discarded each packet whose delay was more than 1 s. Therefore, the average delay of the packet delivered at the sink remained almost constant. This feature also decreased the energy efficiency and the PDR. The MLCL protocol had a lower delay and higher PDR compared with the MMSPEED and QoSMOS protocols. This is because the MLCL protocol estimates the minimum delay with which a packet can be delivered to the sink, but the MMSPEED and QoSMOS protocols use local information to route data to the sink. Also, the MMSPEED protocol uses multi-path routing to increase the reliability. This increases the traffic load and has a negative impact on the end-to-end delay.

Two scenarios were considered for a detailed performance comparison among the MLCL, QoSMOS, and MMSPEED protocols. The effects of the event frequency and the number of events were evaluated in the first and second scenarios, respectively. Two classes A and B were considered in the two scenarios. The end-to-end delay limitations were 0.5 s and 1 s for classes A and B, respectively.

In the first scenario, the network was composed of 50 nodes in a 50 m×50 m area. Four events of both classes were centered at coordinates (5, 5), (5, 45), (45, 2), and (45, 45), and their frequency was 2.5 Hz. The sink was positioned at coordinates (10, 25). The average end-to-end delay versus the event frequency is shown in Fig. 9, and the PDR versus the event frequency in Fig. 10.

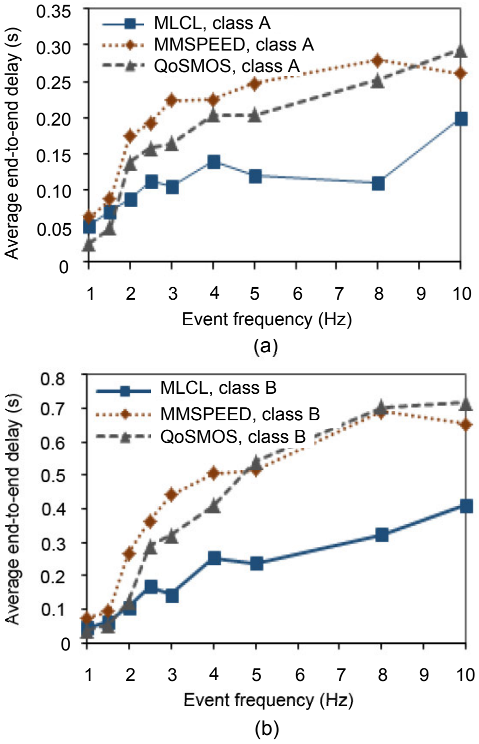

Fig. 9 Comparative average end-to-end delay of class A (a) and class B (b) in the first scenario

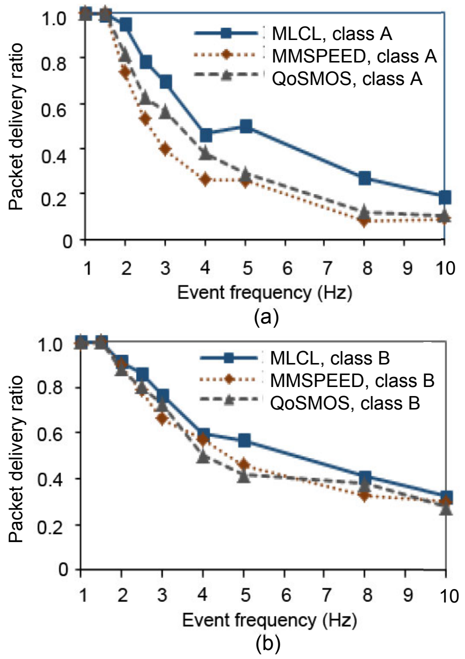

Fig. 10 Comparative packet delivery ratio of class A (a) and class B (b) in the first scenario

For all protocols, it is clear that the average end-to-end delay of class A was less than that of class

B (Fig. 9). Also, the average end-to-end delay of the MLCL protocol was less than that of the MMSPEED and QoSMOS protocols for both classes.

For both classes, the PDR of the MLCL protocol was higher than that of the MMSPEED and QoSMOS protocols, because the MLCL protocol employs the initial phase to route packets with the minimum hop count to the sink. Furthermore, it estimated the MDT, and then the QoS checker dropped a packet whose TTL field was less than this minimum delay. This dropping prevented the transmission of unusable packets to the network. Consequently, both parameters were improved because the congestion probability and collision level were reduced.

In the second scenario, the network was composed of 200 nodes in a 100 m×100 m area. Four events of both classes were centered at coordinates (5, 5), (5, 95), (95, 5), and (95, 95), and the sink was positioned at coordinates (50, 50). To increase the number of events, other events which also had both classes occurred at random coordinates. The average end-to-end delay versus the number of events is shown in Fig. 11, and the PDR versus the number of events in Fig. 12.

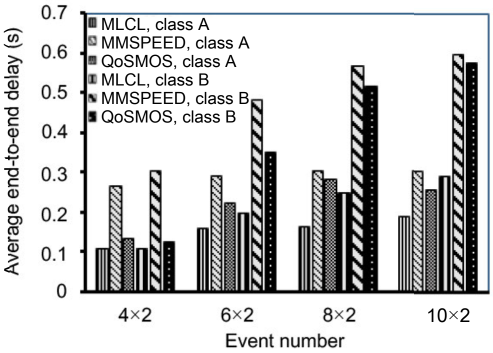

Fig. 11 Comparative average end-to-end delay of classes A and B in the second scenario

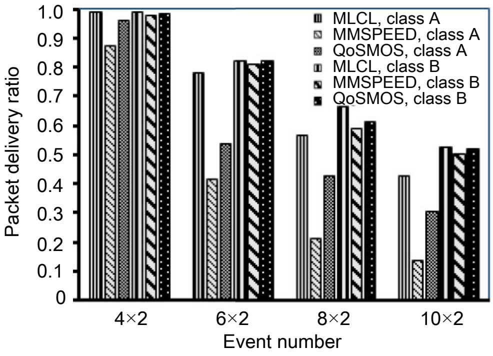

Fig. 12 Comparative packet delivery ratio of classes A and B in the second scenario

The MLCL protocol performed better than the MMSPEED and QoSMOS protocols in terms of end-to-end delay and reliability in large-scale networks (Figs. 11 and 12). When the number of events increased, the traffic load increased. Thus, the collision probability increased and the reaching probability decreased. The MMSPEED protocol uses the multi-path to increase the PDR, but this approach causes collisions and congestion through the network. Thus, the delay of a packet was increased, and then more packets were dropped. Therefore, the PDR was reduced for end-to-end delay provisioning.

6 Conclusions

In this paper, a new architecture is proposed for end-to-end delay provisioning in WMSNs. The MLCL protocol combines the functionality of the MAC and network layers. The MLCL protocol uses the level of each node to route a packet to the sink node. This level causes each packet to be delivered to the sink with a minimum hop count. Thus, the average end-to-end delay is reduced.

The MLCL protocol estimates the minimum delay for a packet to be delivered to the sink. This estimation is used to find a path with the minimum delay. The MLCL protocol uses the TTL updater to calculate the deadline of a packet. The QoS checker uses this deadline to drop an unusable packet. Therefore, it overcomes the congestion and increases the PDR. The results obtained show that the proposed protocols achieve desirable improvements in terms of end-to-end delay and PDR.

Compliance with ethics guidelines

Hossein HADADIAN NEJAD YOUSEFI, Yousef SEIFI KAVIAN, and Alimorad MAHMOUDI declare that they have no conflict of interest.

References

Akyildiz IF, Melodia T, Chowdhury KR, 2006. A survey on wireless multimedia sensor networks. *Comput Netw*, 51(4):921-960.
https://doi.org/10.1016/j.comnet.2006.10.002

Akyildiz IF, Melodia T, Chowdhury KR, 2008. Wireless multimedia sensor networks: applications and testbeds. *Proc IEEE*, 96(10):1588-1605.
https://doi.org/10.1109/JPROC.2008.928756

Al-Wazedi I, Elhakeem AK, 2011. Cross layer design using adaptive spatial TDMA and optimum routing for wireless mesh networks. *AEU Int J Electron Commun*, 65(1):44-52. https://doi.org/10.1016/j.aeue.2010.01.003

Bhuiyan MM, Gondal I, Kamruzzaman J, 2011. CODAR: congestion and delay aware routing to detect time critical events in WSNs. Proc Int Conf on Information Networking, p.357-362. https://doi.org/10.1109/ICOIN.2011.5723128

Dargie W, Poellabauer C, 2010. Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley, Chichester, West Sussex, UK.

Deb B, Bhatnagar S, Nath B, 2003. ReInForM: reliable information forwarding using multiple paths in sensor networks. Proc 28th Annual IEEE Int Conf on Local Computer Networks, p.406-415. https://doi.org/10.1109/LCN.2003.1243166

Demir AK, Demiray HE, Baydere S, 2011. XLCM: xross-layer communication module for service differentiation in wireless sensor networks. Proc 7th Int Wireless Communications and Mobile Computing Conf, p.565-570. https://doi.org/10.1109/IWCMC.2011.5982595

Demir AK, Demiray HE, Baydere S, 2014. QoSMOS: cross-layer QoS architecture for wireless multimedia sensor networks. *Wirel Netw*, 20(4):655-670. https://doi.org/10.1007/s11276-013-0628-3

Ehsan S, Hamdaoui B, 2012. A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks. *IEEE Commun Surv Tutor*, 14(2):265-278. https://doi.org/10.1109/SURV.2011.020211.00058

Felemban E, Lee CG, Ekici E, 2006. MMSPEED: multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Trans Mob Comput*, 5(6):738-754. https://doi.org/10.1109/TMC.2006.79

Feng W, Feng SL, Ding YH, et al., 2014. Cross-layer resource allocation in wireless multi-hop networks with outdated channel state information. *J Zhejiang Univ-Sci C (Comput & Electron)*, 15(5):337-350. https://doi.org/10.1631/jzus.C1300315

Hadadian H, Kavian YS, 2016. Cross-layer protocol using contention mechanism for supporting big data in wireless sensor network. Proc 10th Int Symp on Communication Systems, Networks and Digital Signal Processing, p.1-5. https://doi.org/10.1109/CSNDSP.2016.7573996

Hamid Z, Bashir F, 2013. Xl-WMSN: cross-layer quality of service protocol for wireless multimedia sensor networks. *EURASIP J Wirel Commun Netw*, 2013(1):174. https://doi.org/10.1186/1687-1499-2013-174

Hamid Z, Hussain FB, 2014. QoS in wireless multimedia sensor networks: a layered and cross-layered approach. *Wirel Pers Commun*, 75(1):729-757. https://doi.org/10.1007/s11277-013-1389-0

He T, Stankovic JA, Lu CY, et al., 2003. SPEED: a stateless protocol for real-time communication in sensor networks. Proc 23rd Int Conf on Distributed Computing Systems, p.46-55. https://doi.org/10.1109/ICDCS.2003.1203451

Hou IH, 2015. Packet scheduling for real-time surveillance in multihop wireless sensor networks with lossy channels. *IEEE Trans Wirel Commun*, 14(2):1071-1079. https://doi.org/10.1109/TWC.2014.2363678

Huang GM, Tao WJ, Liu PS, et al., 2013. Multipath ring routing in wireless sensor networks. Proc 2nd Int Symp on Computer, Communication, Control and Automation, p.768-771. https://doi.org/10.2991/isccca.2013.193

Karaca O, Sokullu R, 2012. A cross-layer fault tolerance management module for wireless sensor networks. *J Zhejiang Univ-Sci C (Comput & Electron)*, 13(9):660-673. https://doi.org/10.1631/jzus.C1200029

Kruk Ł, Lehoczky J, Ramanan K, et al., 2011. Heavy traffic analysis for EDF queues with reneging. *Ann Appl Prob*, 21(2):484-545. https://doi.org/10.1214/10-AAP681

Kumar VN, Sankar KS, Rao LS, et al., 2012. Comparative analysis of QoS-aware routing protocols for wireless sensor networks. *Innov Syst Des Eng*, 3(3):100-104.

Li L, Liu YH, Wang J, et al., 2018. Partially observed cross-layer optimization for vehicular communications. *Int J Commun Syst*, 31(1):e3398. https://doi.org/10.1002/dac.3398

Lin XH, Kwok YK, Wang H, 2009. Cross-layer design for energy efficient communication in wireless sensor networks. *Wirel Commun Mob Comput*, 9(2):251-268. https://doi.org/10.1002/wcm.608

Lin ZC, van der Schaar M, 2011. Autonomic and distributed joint routing and power control for delay-sensitive applications in multi-hop wireless networks. *IEEE Trans Wirel Commun*, 10(1):102-113. https://doi.org/10.1109/TWC.2010.111910.091238

Mendes LDP, Rodrigues JJPC, 2011. A survey on cross-layer solutions for wireless sensor networks. *J Netw Comput Appl*, 34(2):523-534. https://doi.org/10.1016/j.jnca.2010.11.009

Messaoudi A, Elkamel R, Helali A, et al., 2017. Cross-layer based routing protocol for wireless sensor networks using a fuzzy logic module. Proc 13th Int Wireless Communications and Mobile Computing Conf, p.764-769. https://doi.org/10.1109/IWCMC.2017.7986381

Misra S, Reisslein M, Xue GL, 2008. A survey of multimedia streaming in wireless sensor networks. *IEEE Commun Surv Tutor*, 10(4):18-39. https://doi.org/10.1109/SURV.2008.080404

Pandya A, Mehta M, 2012. Performance evaluation of multipath ring routing protocol for wireless sensor network. Proc Int Conf on Advances in Computer, Electronics and Electrical Engineering, p.410-414.

Sahoo A, Chilukuri S, 2010. DGRAM: a delay guaranteed routing and MAC protocol for wireless sensor networks. *IEEE Trans Mob Comput*, 9(10):1407-1423. https://doi.org/10.1109/TMC.2010.107

Singh R, Verma AK, 2017. Energy efficient cross layer based adaptive threshold routing protocol for WSN. *AEU Int J Electron Commun*, 72:166-173. https://doi.org/10.1016/j.aeue.2016.12.001

Vuran MC, Akyildiz IF, 2010. XLP: a cross-layer protocol for efficient communication in wireless sensor networks. *IEEE Trans Mob Comput*, 9(11):1578-1591. https://doi.org/10.1109/TMC.2010.125

Wang C, Li B, Sohraby K, et al., 2007. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE J Sel Areas Commun*, 25(4):786-795. https://doi.org/10.1109/JSAC.2007.070514

Wang HG, Peng DM, Wang W, et al., 2008. Cross-layer routing optimization in multirate wireless sensor networks for distributed source coding based applications. *IEEE Trans Wirel Commun*, 7(10):3999-4009. https://doi.org/10.1109/T-WC.2008.070516

Wang HG, Peng DM, Wang W, et al., 2009. Image transmissions with security enhancement based on region and path diversity in wireless sensor networks. *IEEE Trans Wirel Commun*, 8(2):757-765. https://doi.org/10.1109/TWC.2009.070769

Wang W, Peng DM, Wang HG, et al., 2009. Cross-layer multirate interaction with distributed source coding in wireless sensor networks. *IEEE Trans Wirel Commun*, 8(2):787-795. https://doi.org/10.1109/TWC.2009.071009

Yigitel MA, Incel OD, Ersoy C, 2011. QoS-aware MAC protocols for wireless sensor networks: a survey. *Comput Netw*, 55(8):1982-2004. https://doi.org/10.1016/j.comnet.2011.02.007